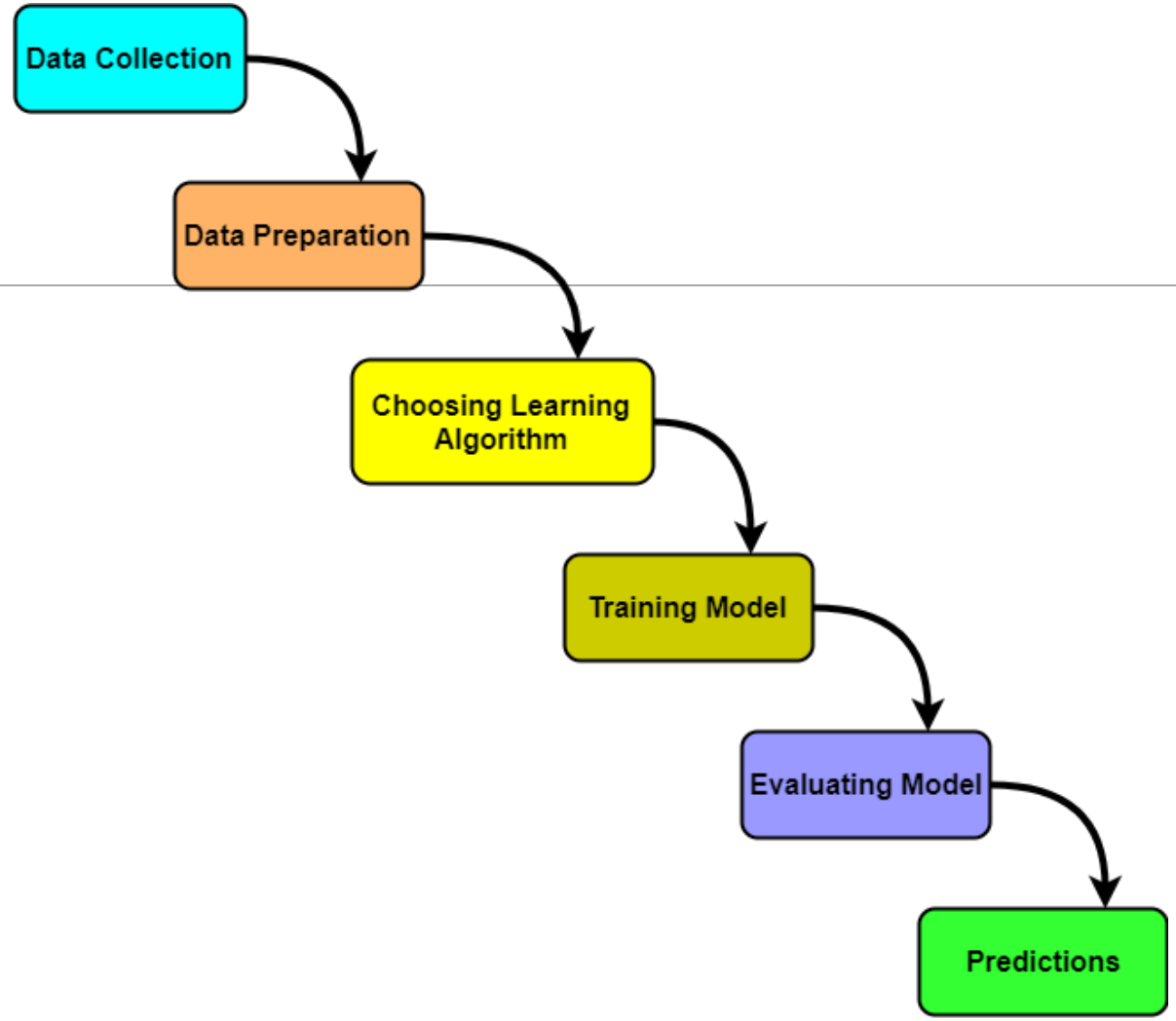
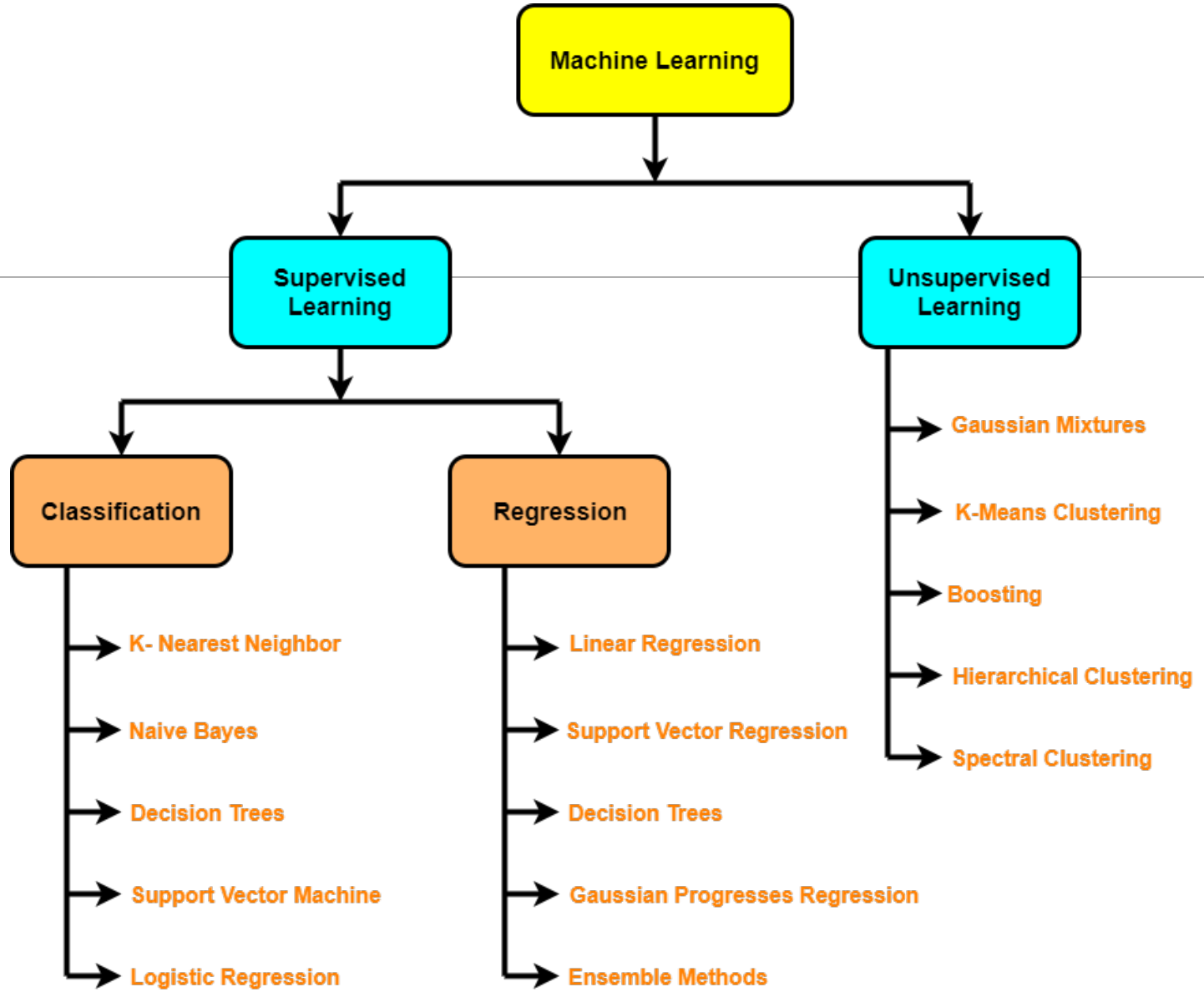
Decision Tree & Random Forest



Machine Learning Workflow



Decision Tree

- Flow chart like structure that helps us make decisions
-
- Family of supervised learning algorithm
-
- Used for solving regression and classification problems
-
- The regression tree is used when the predicted outcome is a real number and
- classification tree is used to predict the class to which the data belongs.

Applications of Decision Tree

- Business Management
- Customer Relationship Management
- Fraudulent Statement Detection
- Energy Consumption
- Healthcare Management
- Fault Diagnosis

Terminology related to Decision Trees

1.Root Node

2.Splitting

3.Decision Node

4.Leaf / Terminal Node

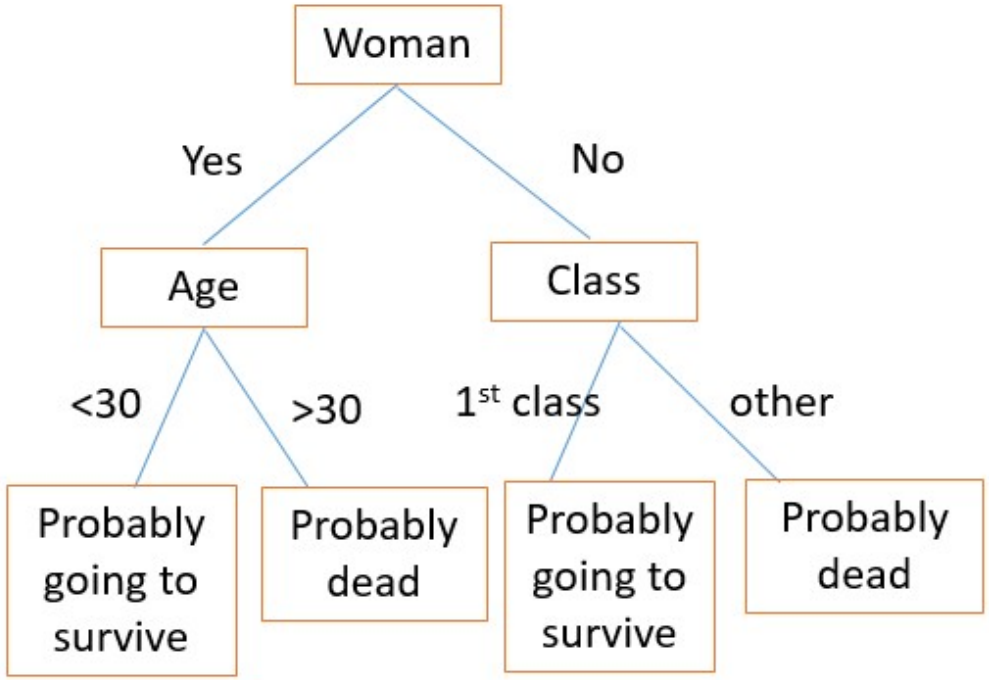
5.Pruning

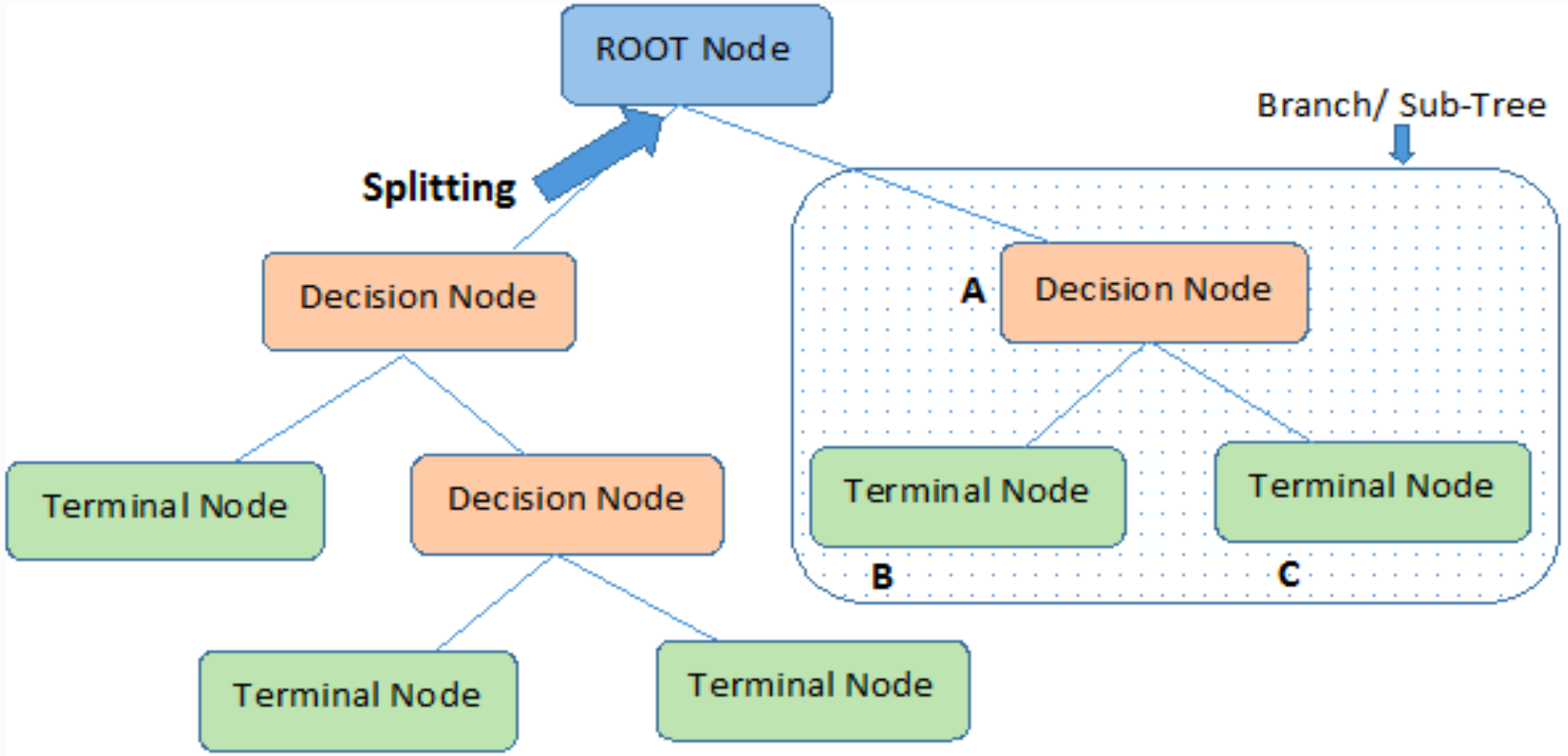
6.Branch / Sub-Tree

7.Parent and Child Node

A normal tree

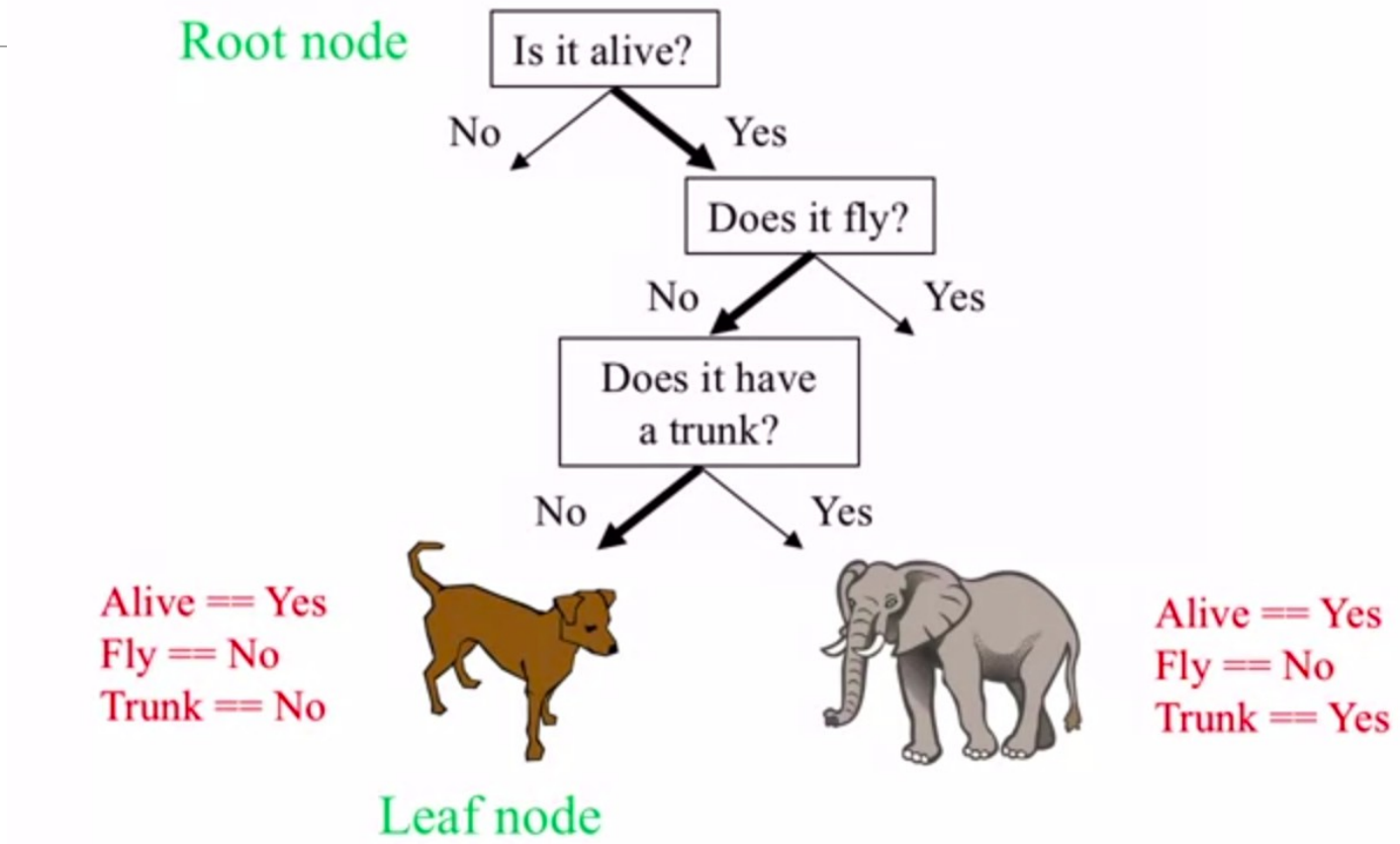
A decision tree

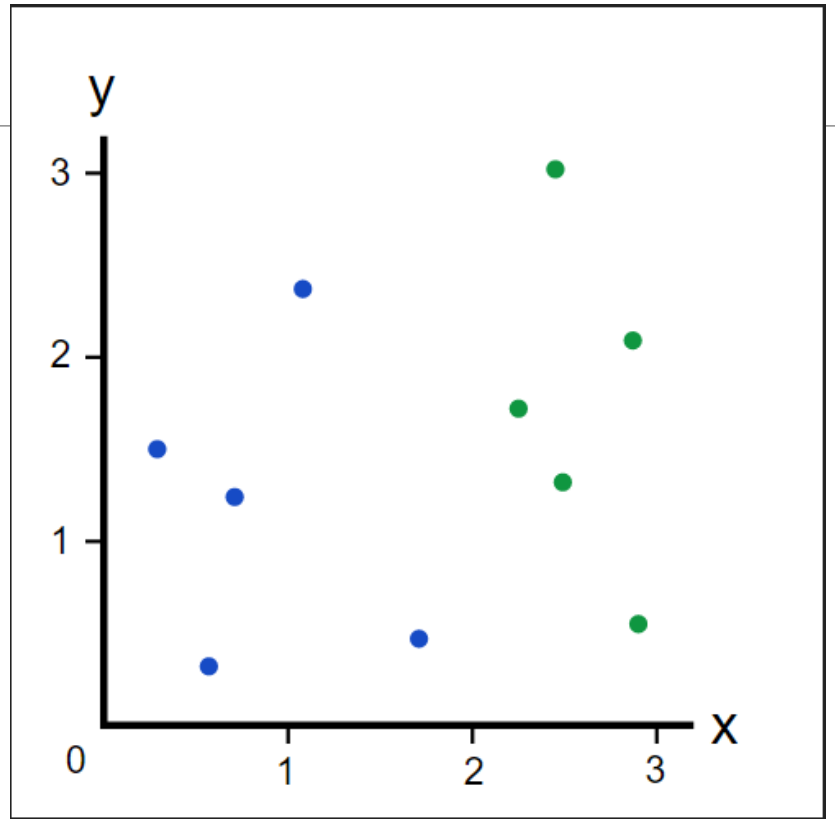




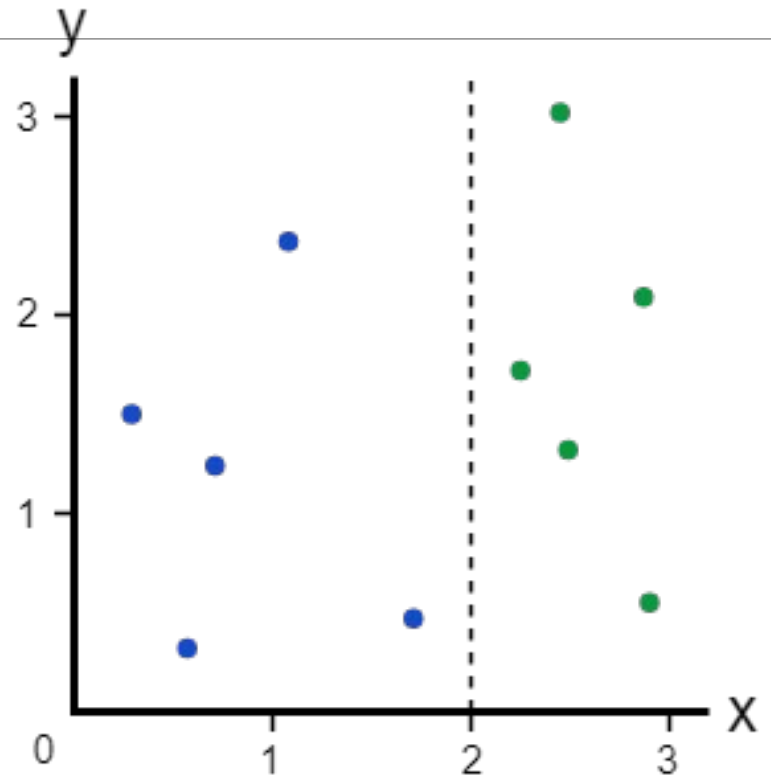
Note:- A is parent node of B and C.

Decision Tree Example

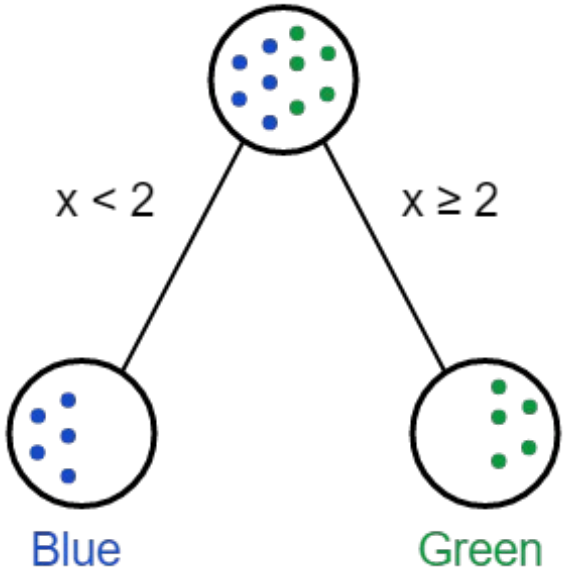




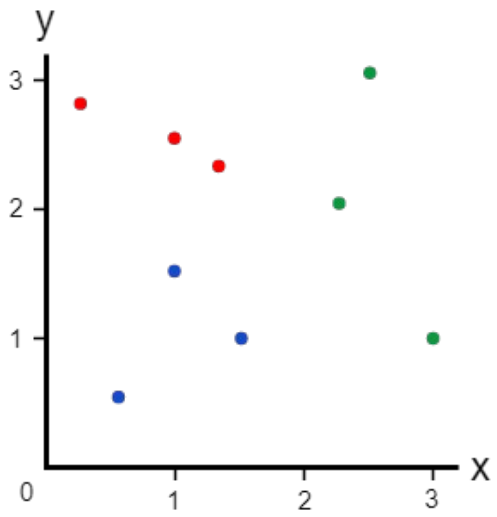
There was a new point with an x coordinate of 1, what colour do you think it'd be?

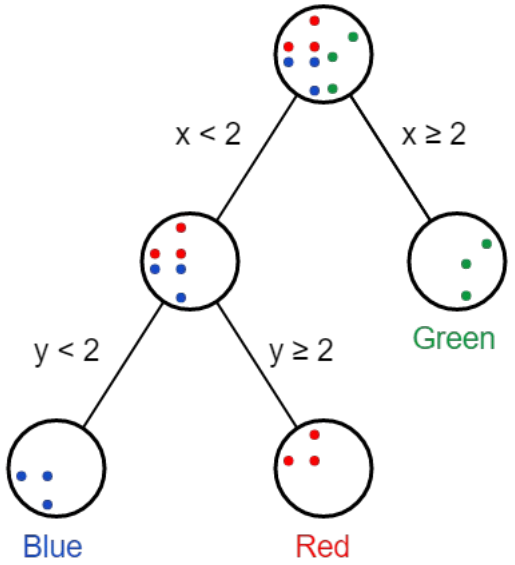


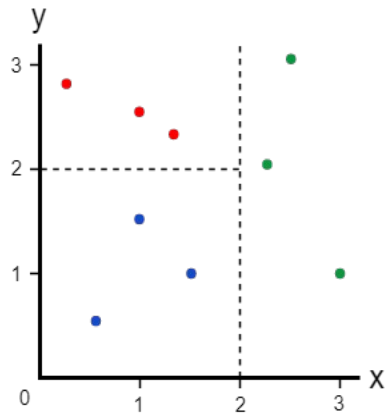
The Dataset, split at x=2



- That’s a simple decision tree with one decision node that tests x < 2.
- If the test passes (x < 2), we take the left branch and pick Blue.
- If the test fails (x≥2), we take the right branch and pick Green.



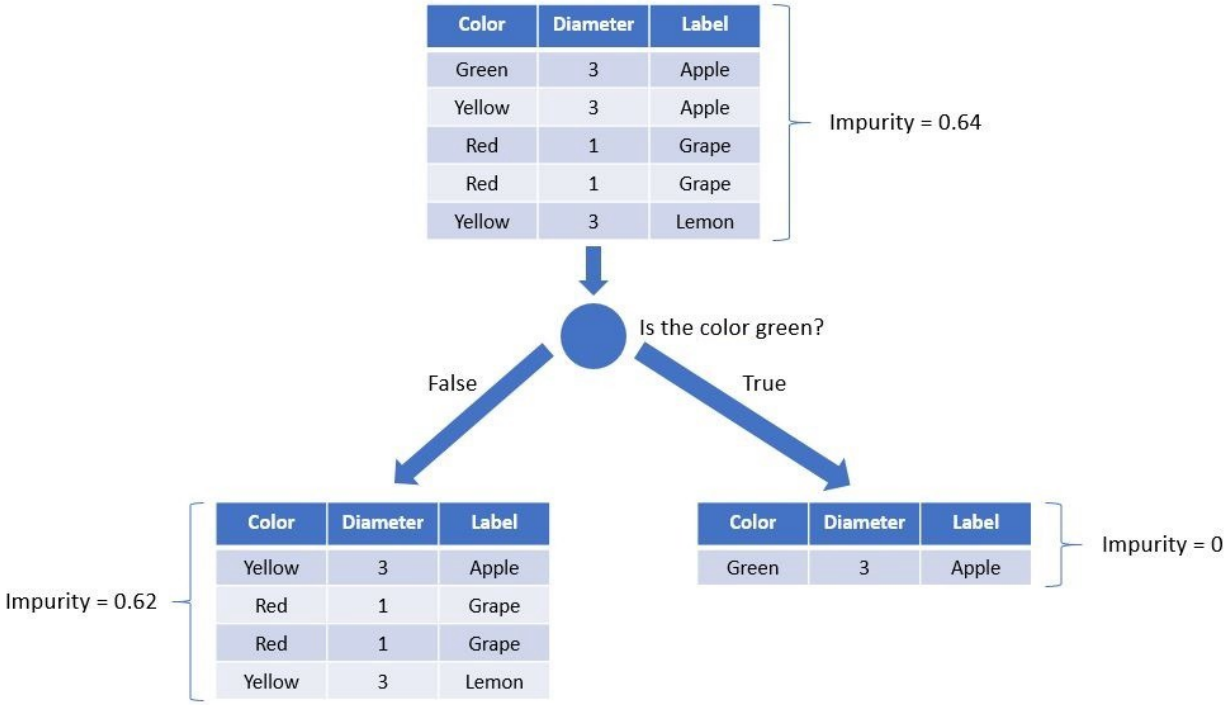




- If $x \geq 2$, we can still confidently classify it as green. ●
- If $x < 2$, we can't immediately classify it as blue - it could be red, too. ● ●

We need to add another decision node to our decision tree

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous).



$$Information\ Gain = 0.64 - \left(\frac{4}{5} * 0.62 + \frac{1}{5} * 0\right) = 0.144$$

Types of Decision Tree Algorithms

- ID3
- C4.5
- CART
- ASSISTANT

ID3

- Ross Quinlan
- Iterative Dichotomiser 3
- Classification algorithm
- evaluate split points:entropy and information gain

Attribute Selection Measures

- Entropy & Information Gain
- Gini Index

Entropy

- ID3 algorithm uses entropy to calculate the homogeneity of a sample.
- If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.
- Entropy values can fall between 0 and 1
- In order to select the best feature to split on and find the optimal decision tree, the attribute with the smallest value of entropy should be used.

A1	A2	Class
-	-	M
-	-	M
-	-	M
-	-	M

- In a sub set all the samples are in the same class, that subset is called a Pure Node.
- Entropy is the measure of impurity.
- For a pure node entropy equal to zero.

$$H(S) = -\sum_{i=1}^{N} P_i log_2 P_i$$

S - Set of all instances

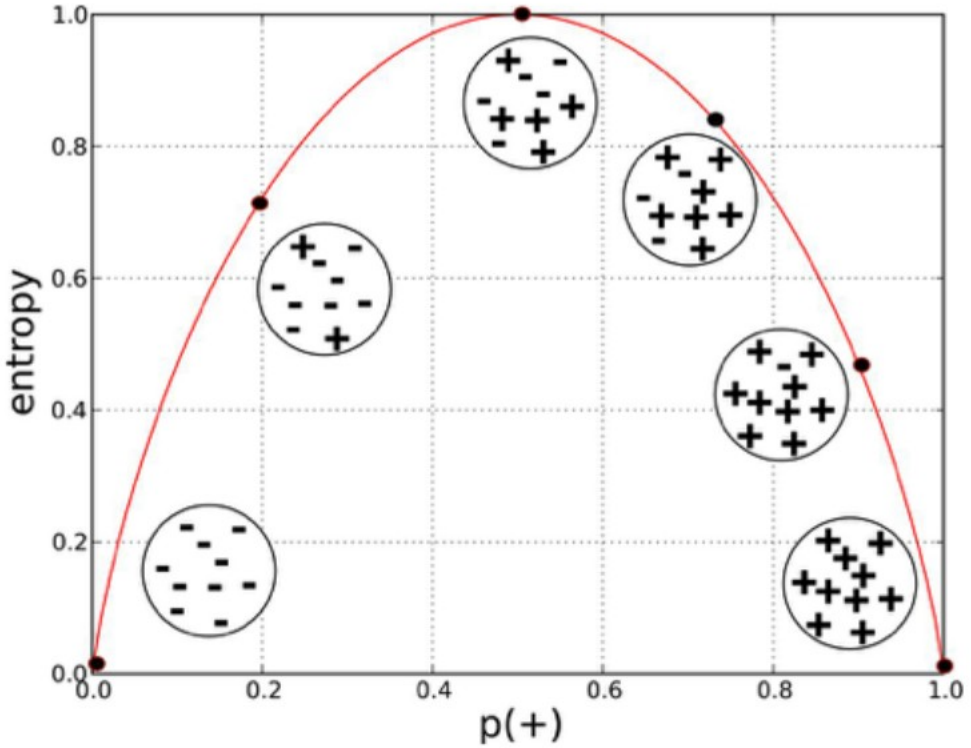
N - Number of distinct class values

Pi - Event probablity

Let’s say we only have two classes ,
a positive class and a negative class.
Therefore ‘i’ here could be either + or - .
So if we had a total of 100 data points in our dataset with 30 belonging to the positive class and 70 belonging to the negative class then
‘P+’ would be 3/10 and ‘P-’ would be 7/10.
entropy of my classes

$$-\frac{3}{10}\times\log_2\left(\frac{3}{10}\right)-\frac{7}{10}\times\log_2\left(\frac{7}{10}\right)\approx 0.88$$

High entropy--high level of disorder (meaning low level of purity)



- **Information Gain**
- Information gain represents the difference in entropy before and after a split on a given attribute.
- The attribute with the highest information gain will produce the best split.

Gini Index

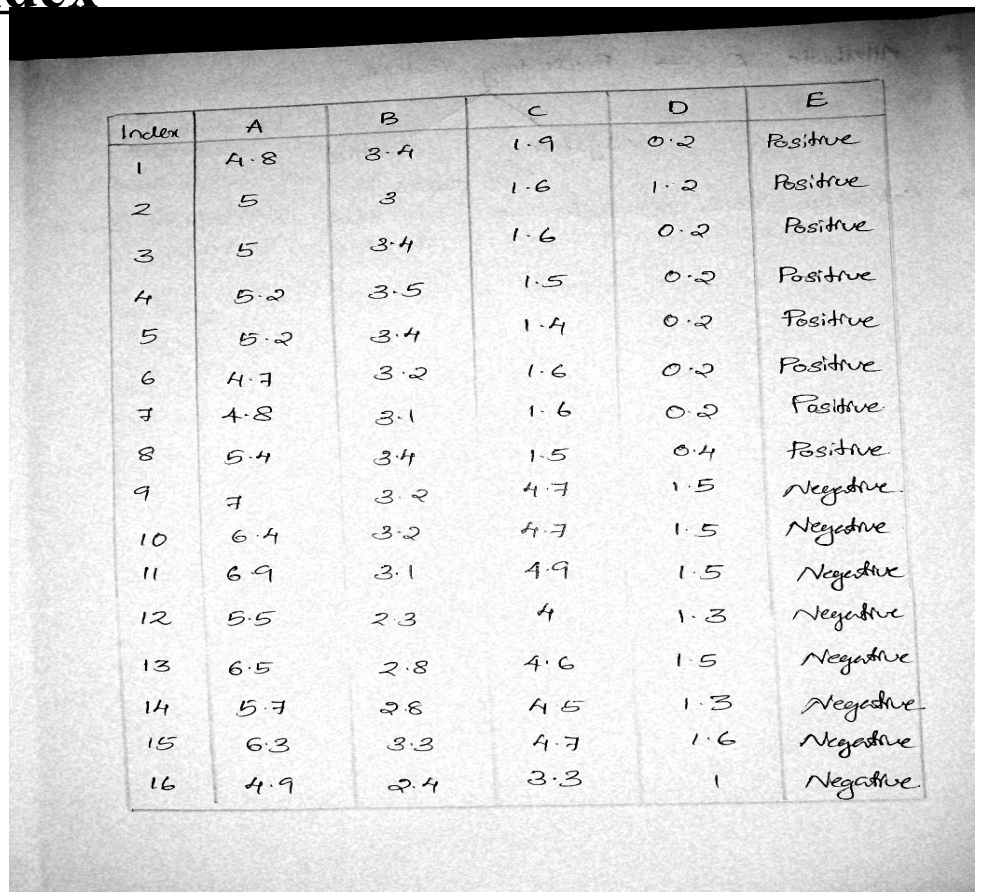
The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optima split from a root node, and subsequent splits.

Gini Impurity is calculated using the formula,

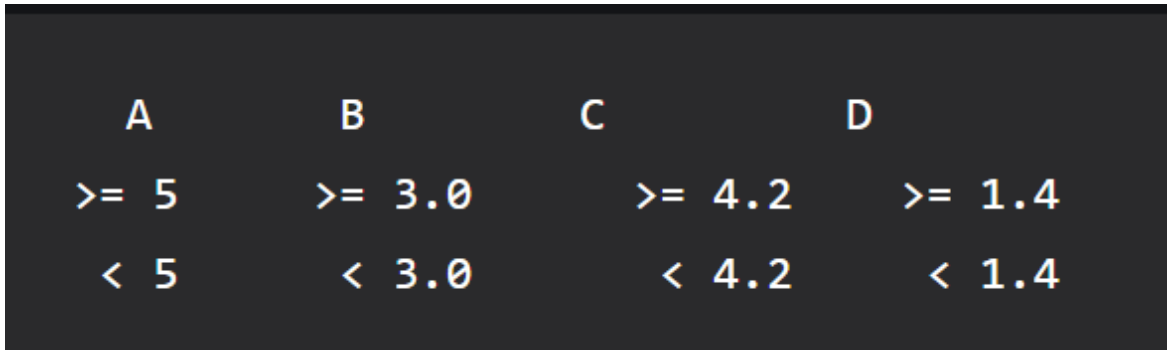
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

- Lower the Gini Impurity, higher is the homogeneity of the node.
- The Gini Impurity of a pure node(same class) is zero.

Decision tree using Gini index

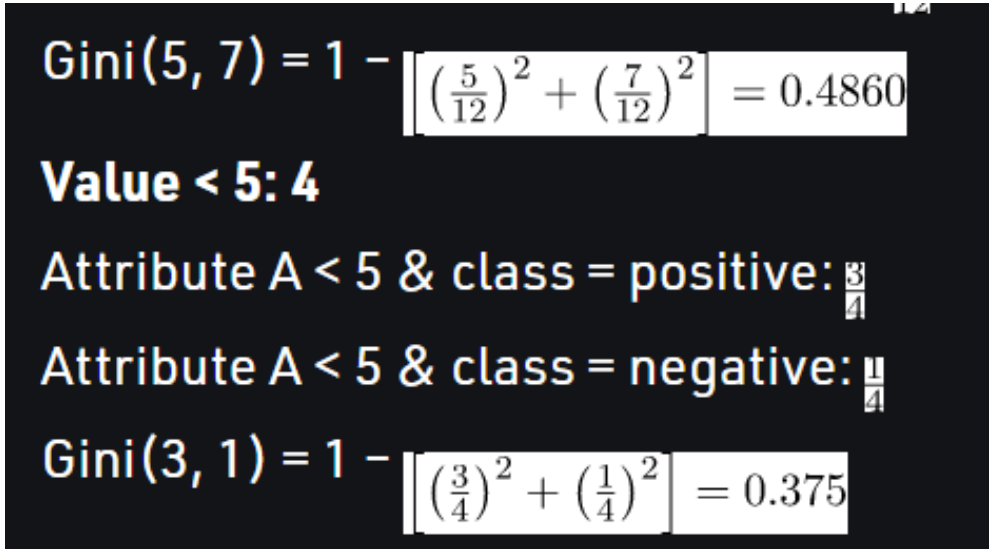


Index	A	B	C	D	E
1	4.8	3.4	1.9	0.2	Positive
2	5	3	1.6	1.2	Positive
3	5	3.4	1.6	0.2	Positive
4	5.2	3.5	1.5	0.2	Positive
5	5.2	3.4	1.4	0.2	Positive
6	4.7	3.2	1.6	0.2	Positive
7	4.8	3.1	1.6	0.2	Positive
8	5.4	3.4	1.5	0.4	Positive
9	7	3.2	4.7	1.5	Negative
10	6.4	3.2	4.7	1.5	Negative
11	6.9	3.1	4.9	1.5	Negative
12	5.5	2.3	4	1.3	Negative
13	6.5	2.8	4.6	1.5	Negative
14	5.7	2.8	4.5	1.3	Negative
15	6.3	3.3	4.7	1.6	Negative
16	4.9	2.4	3.3	1	Negative


A
>= 5
< 5
B
>= 3.0
< 3.0
C
>= 4.2
< 4.2
D
>= 1.4
< 1.4

Calculating Gini Index for Attribute A

Value >= 5: 12
Attribute A >= 5 & class = positive: 5/12
Attribute A >= 5 & class = negative: 7/12



Gini(Target,A)=(12/16)*(0.486)+(4/16)*(0.375)=0.45826

Calculating Gini Index for Var B:

Value >= 3: 12

Attribute B >= 3 & class = positive: $\frac{8}{12}$

Attribute B >= 5 & class = negative: $\frac{4}{12}$

Gini(5, 7) = 1 − $\left[\left(\frac{8}{12}\right)^2 + \left(\frac{4}{12}\right)^2\right] = 0.4460$

Value < 3: 4

Attribute A < 3 & class = positive: $\frac{0}{4}$

Attribute A < 3 & class = negative: $\frac{4}{4}$

Gini(3, 1) = 1 − $\left[\left(\frac{0}{4}\right)^2 + \left(\frac{4}{4}\right)^2\right] = 1$

By adding weight and sum each of the gini indices:

$$gini(Target, B) = \left(\frac{12}{16}\right) * (0.446) + \left(\frac{0}{16}\right) * (0) = 0.3345$$

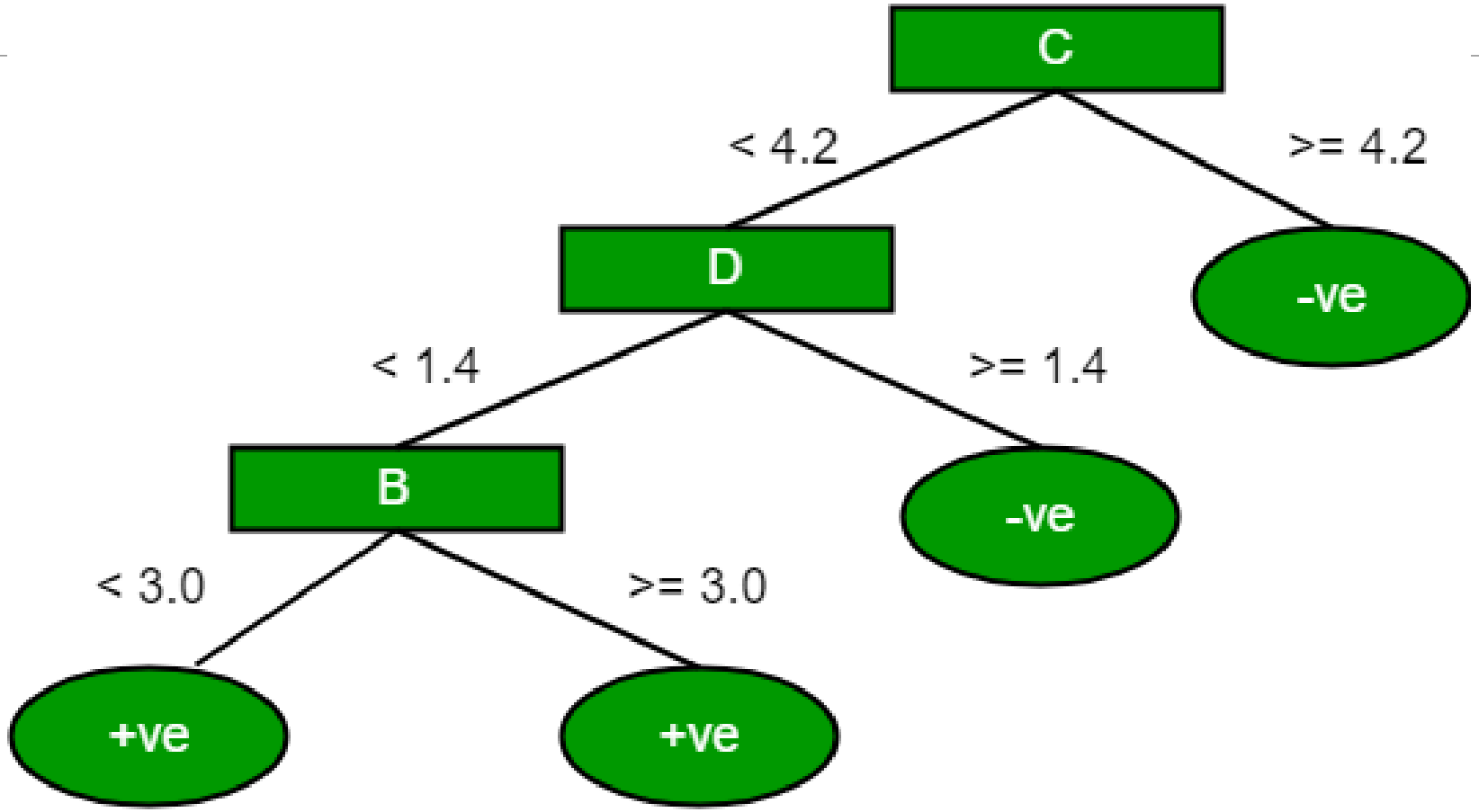
```
            Positive    Negative
For A|>= 5.0    5       7
     |<5    3       1
Ginin Index of A = 0.45825
```

```
            Positive    Negative
For B|>= 3.0    8       4
     |< 3.0    0       4
Gini Index of B= 0.3345
```

```
            Positive    Negative
For C|>= 4.2    0       6
     |< 4.2    8       2
Gini Index of C= 0.2
```

```
             Positive    Negative
For D|>= 1.4    0       5
     |< 1.4    8       3
Gini Index of D= 0.273
```

Decision tree for above dataset



Inclusys Org Foundation
Empowering the Neurodivergent

Tree Pruning

- Pruning is when you selectively remove branches from a tree.
- The goal is to remove unwanted branches, improve the tree's structure, and direct new, healthy growth.

- When decision trees are built, many of the branches may reflect noise or outliers in the training data.
- Tree pruning methods address this problem of overfitting the data.
- Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy.

Types of tree pruning

- Post Pruning
- Pre-Pruning

Pre Pruning

- used before construction of decision tree
- Pre-Pruning can be done using Hyperparameter tuning
- Overcome the overfitting issue

Pre-pruning techniques:

Max depth:

It is used to pass maximum depth of a tree i.e. length of longest path from a root node to leaf node. By passing this parameter, we can stop the growth of the tree.

Minsplit:

It is the minimum number of records that must exist in a node to split a node.

Minbucket:

It is the minimum number of records that can be present in a terminal node.

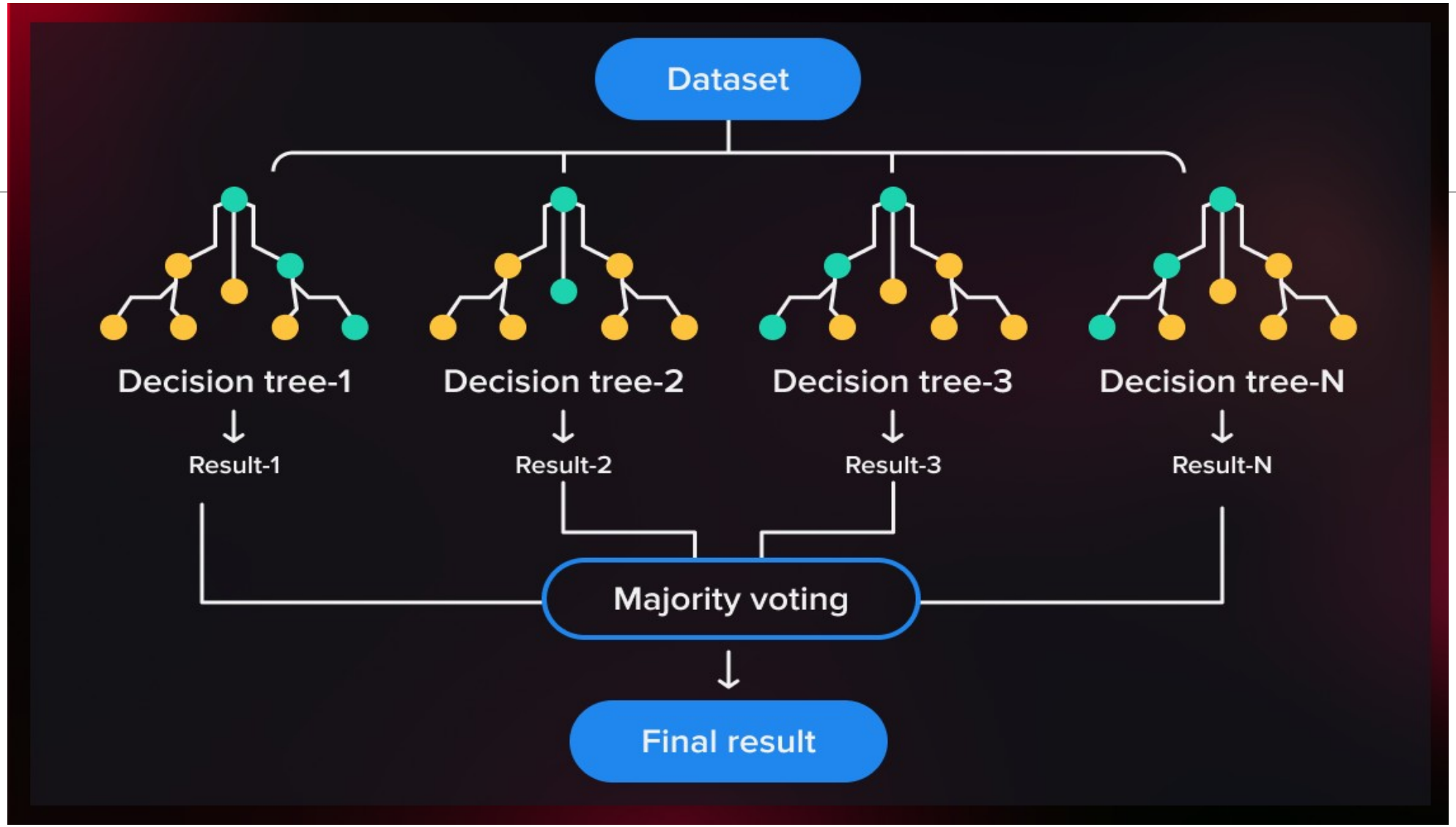
Post Pruning

- used after construction of decision tree
- used when decision tree will have very large depth and will show overfitting of model(used when we have infinitely grown decision tree)
- It is also known as backward pruning

Random Forest

- Belongs to the supervised learning technique.
- used for both Classification and Regression problems in ML
- It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset

- Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Key Differences:

- Ensemble Learning: Random Forest is an ensemble learning method, which means it combines multiple decision trees to make a final prediction. On the other hand, Decision Tree is a single tree model.
- Tree Generation: In Random Forest, multiple trees are generated from different random samples of the training data. In Decision Tree, a single tree is built using all the training data.
- Feature Selection: In Random Forest, each tree uses a random subset of features to split nodes. This results in a reduction of overfitting compared to a single Decision Tree.

- Prediction: In Random Forest, the final prediction is made by aggregating the predictions of all the trees, either by voting for classification tasks or by taking the average for regression tasks. In Decision Tree, the prediction is made by following the path through the tree that leads to the final prediction.

- Bias-Variance Trade-off: Decision Trees have a tendency to overfit the training data, leading to high variance and low bias. Random Forest, on the other hand, has low variance and high bias because of the aggregation of multiple trees.

Inclusys Org Foundation
Empowering the Neurodivergent

Inclusys Org Foundation
Empowering the Neurodivergent

Thank You